最適符号探索のための FPGA ベースヘテロジニアスアクセラレータに関する研究

修士学位論文要約（平成31年３月）

最適符号探索のための FPGA ベース ヘテロジニアスアクセラレータに関する研究

平舘　侑樹
指導教員：張山　昌論

Study on FPGA Based Heterogeneous Accelerator for Optimum Code Search

Yuki Hiradate
Supervisor: Masanori HARIYAMA

Optimum code search refer to searching a particular bit pattern that satisfies given constraints. Obtaining such codes is very important in fields such as error correcting, cryptography, etc. Unfortunately, the search time increases exponentially with the number of bits in the code, and typically requires many months of computation to find large codes. In this paper, we propose a heterogeneous system with a CPU and an FPGA to speed-up optimum code search. According to the evaluation, we obtain about 100 times speed-up compared to typical CPU-based implementation for Extremal Doubly Even Self-Dual code of length 128 search problem. And we could verify we can not compose it with 23-out-of-64 code.

1.　はじめに

最適符号探索とは、特定の条件を満たす符号を見つけることを目的とする問題の事である。例として、線形の誤り訂正符号で、誤り訂正能力の高い符号のための最小ハミングウェイトが最大となる符号の探索などが挙げられる。

本研究では、最適符号探索の中で、Extremal Doubly Even Self-Dual code(以下、EDS code と略す) を対象とする。EDS code は、最小ハミングウェイトが大きいという利点があり、誤り訂正や暗号などに応用することができる。実際に、長さ 24 の EDS code は Golay code とよばれ、1980 年頃のボイジャー宇宙計画の際に、カラー画像を転送する際に用いられた [1]。符号長が n である EDS code は以下の 3 つの条件を満たす符号のことを言う。

- ハミングウェイトが 4 の倍数
- 自己双対性を持つ
- 最小ハミングウェイトが $4\lfloor\frac{n}{24}\rfloor+4$

EDS code の構成法は原田の研究 [2] で提案されている。長さ 128 の EDS code の構成法をまとめると以下の 3 つのステップからなる。ここで生成する符号を C とする。

step 1　2 進数で符号長が 64bit かつハミングウェイトが 3 (mod 4) である符号 x を生成
step 2　$\boldsymbol{AA^T}+\boldsymbol{BB^T}=\boldsymbol{I}$ となるかを判定
step 3　C の生成行列 G の 10 行目まで、$C^{\perp}$ の生成行列 H の 9 行目までの和のハミングウェイトが 24 以上であることを調べる

$A(B)$:x の上位 (下位)32bit を 1 列目とする巡回行列

$$G=\begin{pmatrix} I_{64} & A & B \\ & B^{\mathrm{T}} & A^{\mathrm{T}} \end{pmatrix}$$

EDS code 探索には、莫大な処理時間が必要であるという問題点がある。長さが 128 のとき、2^{64} 個の符号を判定する必要があり、1 並列だと 3GHz の CPU で約 194 年もの時間が必要となる。

そのため、本研究では、FPGA と CPU の異種環境を用いることで高並列に条件判定を行うことで処理時間を削減するヘテロジニアスアクセラレータの構成と EDS code の探索を行うことを目的とする。

CPU と FPGA は得意とする演算が異なるため、それぞれが得意な演算のみを実行するようにシステムを構成することで探索の高速化が可能である。

2.　提案アルゴリズム

高速化のために重要なことは無駄な処理をせず、演算数を減らすことである。また、FPGA で高速化のために重要な要素は、並列化とパイプライン化の二つである。そのため、各 step に対して演算量を減らし、高並列化とパイプライン化を可能とするアルゴリズムを提案する。

step 1 では、特定のハミングウェイトを持つ k-out-of-n code を円順列を用いて生成することで、演算数を $\frac{1}{4}$ に減らし、並列にハミングウェイトが 3 (mod 4) である符号を生成する。

step 2 では、行列の性質に着目し、冗長な演算や、答えが既知である演算を省略することで演算数を $\frac{1}{64}$

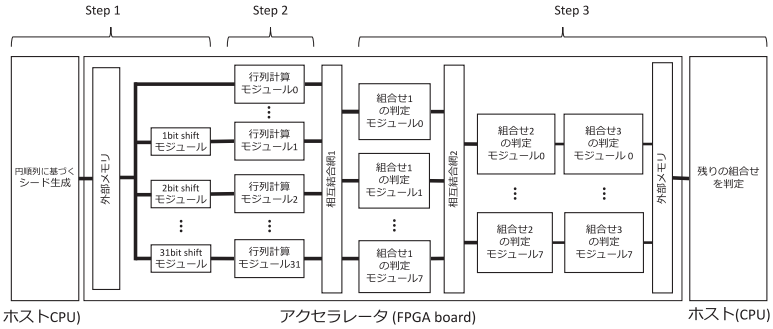

図 1．ヘテロジニアスアクセラレータの構成

に減らす。さらに、パイプライン処理をすることで演算時間を減らす。

step 3 では、条件により判定の中断と、判定する順番を変更、ハミングウェイトが等しくなる組み合わせの判定を行わないことで、演算数を $\frac{1}{5000\text{ 万}}$ に減らす。また、判定する組合せを複数モジュールで分割して、パイプライン処理や判定を並列に行うことで、高速に判定可能とする。

3. システム構成

システム構成は、図 1 に示すように、FPGA でシードから k-out-of-n code を生成し、条件判定を行い、CPU で円順列に基づくシード生成と、残りの判定を行う構成となっている。

step1 では、円順列の基づくシード生成を行い、シードを巡回シフトすることで k-out-of-n code を生成する。シード生成は、直列的で、条件分岐を数多く行うため、周波数が高い CPU に向いている。シフト演算は、単純な演算であり、高並列で処理が可能であるため FPGA で実装する。

step2 の行列計算も、FPGA の豊富にあるロジックエレメントにより、パイプライン処理することが可能である。そのため、FPGA で実装する。

step3 の最小ハミングウェイトの判定では、FPGA で高速に判定可能で、リソース消費量も少ない特定の組み合わせのみを FPGA で判定する。残りの判定は、FPGA で実装すると、リソース消費量に対して、演算速度が低いなどのデメリットが存在するため、CPU で判定を行う。

4. システムの評価

提案システムの評価のために、提案システムと CPU のみで EDS code 探索を行うときの処理時間との比較を行う。比較する CPU のみの探索アルゴリズムとして第 2 節で提案したアルゴリズムを用いて、処理時間を比較する。実行環境として、CPU は Intel(R) Xeon(R) CPU E5-2643@ 3.30GHz を使用し、FPGA は DE5a-Net Arria 10 を使用する。FPGA の周波数は 304[MHz] であった。CPU コンパイラは、Intel コンパイラ 17.0 を使用し、OpenCL コンパイラは aocl 16.1 を使用する。2.1×10^9 個の円順列に対して探索を行った結果を、表 1 に示す。表 1 を見ると、提案システムが最も高速で、CPU のみと比較すると約 100 倍高速であることがわかる。

表 1．提案システムと CPU の比較

	提案システム	CPU のみ
処理時間 [sec]	7.113	702.65

また、本研究では 23-out-of-64 code を生成し EDS code の探索を行った。その結果を表 2 に示す。使用システムは、前述した CPU を 1 つ、前述した FPGA を 2 つ用いて構成した。

表 2．23-out-of-64 code 探索の結果

探索時間	90 日と 8 時間
シードの数	2.293×10^{15}
FPGA の出力数	143

探索は約 90 日で終了し、その結果、23-out-of-64 code では、EDS code を構成することができないことがわかった。

5. まとめ

本研究では、CPU と FPGA を用いたヘテロジニアスアクセラレータを構成した。その結果、CPU のみの場合と比べて提案手法は約 100 倍の速度で探索可能である。また、23-out-of-64 code の探索は約 90 日で終了した。その結果、23-out-of-64 code では EDS code を構成できないことがわかった。

今後の展望は、23-out-of-64 code 以外の探索をすることと、FPGA 内でビット反転などのエラーの検査を行う機能を実装し、信頼度を高めることが挙げられる。

文献

1) 楫 勇一、“OHM 大学テキスト　情報・符号理論” 株式会社 オーム社, 2013
2) Masaaki Harada、“An Extremal Doubly Even Self-Dual Code of Length 112” Electronic J. Combin. 15 (2008), #N33 (5 pp.)